CLAIMS

What is claimed is:

1. A method for dithering color in a graphics system that displays a group of pixels and wherein the color of the pixels is represented by color shades having fewer than eight bits, comprising the steps of:

(a) generating an eight bit color shade value for each pixel representing a desired color for each pixel;

(b) truncating the desired eight bit color shade value to obtain a truncated color shade value;

(c) generating a FRAC value for each pixel from the truncated bits of said eight bit color shade value;

(d) producing a ramp value for each pixel using said FRAC value, wherein said ramp value encodes a discrepancy between the desired eight bit color shade value and the truncated color shade value; and

(e) using a bit from said ramp value to select a color shade value of fewer than eight bits that determines the color of each pixel.

2. The method of claim 1, wherein said truncated bits in step (c) includes fewer than the two least significant bits of said desired eight bit color shade value.

3. The method of claim 2, wherein the truncated bits includes the three least significant bits of said desired eight bit color shade value.

4. The method of claim 2, wherein the step of using a bit from said ramp value to select a color shade value of fewer then eight bits (step e) includes using a value from a look-up table to select said bit from said ramp value.

5. The method of claim 4, wherein each pixel has an x address and a y address and said value from said look-up table is determined from the x address and the y address of the pixel to be rendered.

Sub. A2

6. A method for dithering pixel color in a graphics system that displays a group of pixels in which primary pixel colors are represented by color shades having fewer than eight bits comprising the steps of:

(a) generating an eight bit color shade value for each pixel representing a desired color for each pixel;

(b) truncating the desired eight bit color shade value to produce a first color shade value comprising fewer than eight bits;

(c) generating a FRAC value for each pixel representing the truncated bits of said desired eight bit color shade value;

33

(d) producing a ramp value for each pixel using said FRAC value, wherein said ramp value encodes a discrepancy between the desired eight-bit color shade value and the first color shade value;

(e) producing an addend value for incrementing said first color shade value;

(f) incrementing said first color shade value by said addend value to produce a second color shade value; and

(g) selecting said first color shade value or said second color shade value to determine the color of each pixel in said group of pixels.

7. The method of claim 6, wherein said step of producing a ramp value (step d) includes producing a ramp value that includes a number of logic one values indicative of said discrepancy between the desired eight bit color shade value and the first color shade value.

8. The method of claim 6, wherein said step of selecting said first color shade value or said second color shade value (step g) is performed in response to the state of a bit from said ramp value.

9. The method of claim 8, wherein each pixel has an x address and a y address and said x address and said y address of a pixel to be rendered are used to obtain a value from a look-up table, said look-up table value used to select said bit from said ramp value.

34

10. The method of claim 6, wherein said step of incrementing said first color shade (step f) produces on overflow signal if an overflow condition is present.

11. The method of claim 10, wherein said step of selecting said first color shade value or said second color shade value (step g) is performed in response to said overflow signal.

12. A graphics system that displays color shades based upon binary representation having fewer than eight bits, wherein said graphics system initially receives a desired eight bit binary representation for each color shade that is used by the graphics system to render pixels in a pixel grid, said desired eight bit binary representation including upper order bits and lower order bits, comprising:

select fractional logic that receives the desired eight bit binary representation and wherein said select fractional logic produces on its output lines the lower order bits of said desired eight bit binary representation value;

a look-up table that produces a control value based upon an address of each pixel; and

ramp probability logic coupled to said select fractional logic and said look-up table, said ramp probability logic producing a ramp value that encodes a discrepancy between said desired eight bit binary representation and said binary representations having fewer than eight bits.

13. The graphics system of claim 12, further including an addend generator that produces an addend value for incrementing said binary representations having fewer than eight bits.

14. The graphics system of claim 13, further including add logic for producing the sum of said addend value and said binary representations having fewer than eight bits.

15. The graphics system of claim 14, further including a first multiplexer for selecting a bit from said RAMP value, and wherein the bit selection is controlled by said control value produced from said look-up table.

16. The graphics system of claim 15, further including a second multiplexer to which said binary representation having fewer than eight bits and said sum are provided as input signals, and wherein said second multiplexer selects one of a said input signals, said input signal selection controlled by a control signal and said control signal determined by said ramp value.

17. The graphics system of claim 12, wherein said ramp value includes a number of logic 1 values indicative of the discrepancy between said desired eight bit binary representation and said binary representations having fewer than eight bits.

36

18. The graphics system of claim 17, wherein said graphics system represents color using five bits for red and five bits for blue.

19. The graphics system of claim 18, wherein said graphics system represents color using six bits for green.

20. The graphics system of claim 15, wherein said add logic produces an overflow output signal upon detection of an overflow condition.

21. The graphics system of claim 20, wherein said control signal is also determined by said overflow signal.

22. A computer readable storage medium for storing an executable set of software instructions which, when inserted into a host computer system, is capable of controlling the operation of the host computer, said software instructions being operable to dither pixel colors in a graphics system and wherein the color of the pixels is represented by color values having fewer than eight bits, said software instructions including:

means for determining a first index value to a look-up table;

means for providing a look-up table value from said look-up table based on said first index value;

means for determining a ramp probability value; and

means for using said ramp probability value and said look-up table value to determine a dither color value in said graphics system.

23. The invention of claim 22 further including a second index value and wherein said look-up table value is based on said first and said second index values.

24. The invention of claim 23 wherein each pixel on said screen includes a pixel address and said first and said second index values are based on pixel addresses.

25. The invention of claim 24 wherein said pixel addresses comprise a plurality of higher order bits and lower order bits, the lower order bits determining said first and said second index values.

26. The invention of claim 25 wherein the lower order bits used to determine said index values include the least significant three bits of said pixel address.

27. The invention of claim 26 wherein said least significant three bits are obtained by ANDing said pixel address with a value of 7.

28. The invention of claim 27 wherein said pixel address is added to an offset value before ANDing with 7.

29. The invention of claim 28 wherein said look-up table comprises a plurality of rows of binary values, each binary value comprising a plurality of bits wherein said first index value is used to select a row from said look-up table, and wherein said look-up table value is determined by bit shifting said second index value by two bit positions to the left to produce a first bit shifted value and bit shifting said selected row to the right by a number of bit positions equal to said first bit shifted value thereby producing a second bit shifted value.

30. The invention of claim 29 wherein said look-up table value is further determined by selecting the least significant three bits of said second bit shifted value.

31. The invention of claim 30 wherein said color values having fewer than eight bits are generated by truncating eight bit color values and wherein said ramp probability value is determined by computing 2^{FRAC}-1 where FRAC is generated from the truncated bits of said eight bit color values.

32. The invention of claim 31 wherein said dither color value is determined by right bit shifting said ramp probability value by a number of bit positions equal to said look-up table value and then selecting the least significant bit of said bit shifted ramp probability value and setting all other bits in said ramp probability value to zero.

39

33. The invention of claim 32 wherein said bit shifted ramp probability value is left bit shifted to provide an addend value for dithering.

34. The invention of claim 33 wherein said dither value is further determined by adding said addend value to a color value.

35. The invention of claim 34 wherein said added value is added to a color value and if an overflow condition results from said addition, said dither value is set to equal said color value.

sub A5

36. A method for dithering color in a graphics system that displays a group of pixels on a screen, wherein the color of the pixels is represented by color values having fewer than eight bits, said method comprising:

determining a first index value to a look-up table;

determining a look-up table value from said look-up table based on said first index value;

determining a ramp probability value; and

using said ramp probability value and said look-up table value to determine a dither color value in said graphics system.

37. The method of claim 36 further determining a second index value and wherein said look-

40

up table value is also based on said first and said second index values.

38. The method of claim 37 wherein each pixel on said screen includes a pixel address and said first and said second index values are determined based on pixel addresses.

39. The method of claim 38 wherein said pixel addresses comprise a plurality of higher order bits and lower order bits, the lower order bits determining said first and said second index values.

40. The method of claim 39 wherein the lower order bits used to determine said index values include the least significant three bits of said pixel address.

41. The method of claim 40 wherein said least significant three bits are obtained by ANDing said pixel address with a value of 7.

42. The method of claim 41 wherein said pixel address is added to an offset value before ANDing with 7.

43. The method of claim 42 wherein said look-up table comprises a plurality of rows of binary values, each binary value comprising a plurality of bits wherein said first index value is used to select a row from said look-up table, and wherein said look-up table

41

value is determined by bit shifting said second index value by two bit positions to the left to produce a first bit shifted value and bit shifting said selected row to the right by a number of bit positions equal to said first bit shifted value thereby producing a second bit shifted value.

44. The method of claim 43 wherein said look-up table value is further determined by selecting the least significant three bits of said second bit shifted value.

45. The method of claim 44 wherein said color values having fewer than eight bits are generated by truncating eight bit color values and wherein said ramp probability value is determined by computing 2^{FRAC}-1 where FRAC is generated from the truncated bits of said eight bit color values.

46. The method of claim 45 wherein said dither color value is determined by right bit shifting said ramp probability value by a number of bit positions equal to said look-up table value and then selecting the least significant bit of said bit shifted ramp probability value and setting all other bits in said ramp probability value to zero.

47. The method of claim 46 wherein said bit shifted ramp probability value is left bit shifted to provide an addend value for dithering.

42

48. The method of claim 47 wherein said dither value is further determined by adding said addend value to a color value.

49. The method of claim 48 wherein said added value is added to a color value and if an overflow condition results from said addition, said dither value is set to equal said color value.